

# Quelques conclusions sur la construction d'un mailleur 2D de type Delaunay

Houman Borouchaki, Paul-Louis George

## ► To cite this version:

Houman Borouchaki, Paul-Louis George. Quelques conclusions sur la construction d'un mailleur 2D de type Delaunay. [Rapport de recherche] RR-2463, INRIA. 1995. inria-00074212

**HAL Id: inria-00074212**

**<https://hal.inria.fr/inria-00074212>**

Submitted on 24 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

*Quelques conclusions sur la  
construction d'un mailleur 2D de  
type Delaunay*

Houman BOROUCHAKI - Paul-Louis GEORGE

N° 2463

Janvier 1995

PROGRAMME 6

*R*  
*apport  
de recherche*

Les rapports de recherche de l'INRIA  
sont disponibles en format postscript sous  
ftp.inria.fr (192.93.2.54)

si vous n'avez pas d'accès ftp  
la forme papier peut être commandée par mail :  
e-mail : dif.gesdif@inria.fr  
(n'oubliez pas de mentionner votre adresse postale).

par courrier :  
Centre de Diffusion  
INRIA  
BP 105 - 78153 Le Chesnay Cedex (FRANCE)

INRIA research reports  
are available in postscript format  
ftp.inria.fr (192.93.2.54)

if you haven't access by ftp  
we recommend ordering them by e-mail :  
e-mail : dif.gesdif@inria.fr  
(don't forget to mention your postal address).

by mail :  
Centre de Diffusion  
INRIA  
BP 105 - 78153 Le Chesnay Cedex (FRANCE)

# Quelques conclusions sur la construction d'un mailleur 2D de type Delaunay

Houman BOROUCHE et Paul Louis GEORGE

INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex, France.

## Résumé

*Ce papier se propose de faire le point sur les différentes phases nécessaires à la mise en œuvre d'un mailleur automatique de type Delaunay. Il rappelle la méthode et en décrit une variante numériquement robuste en mentionnant à la fois les problèmes à résoudre et les différentes solutions possibles. Seul le cas bidimensionnel est discuté et les extensions possibles au cas tridimensionnel sont indiquées.*

## Conclusion about 2D Delaunay mesh generation

## Abstract

*This paper aims to outline the different phases necessary to implement a Delaunay type automatic mesh generator. It, firstly, summarizes this method and describes a variant which is numerically robust by mentioning at the same time the problems to solve and the different solutions possible. The two-dimensional situation is described fully and possible extensions to the three-dimensional case are briefly indicated.*



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Quelques outils</b>	<b>5</b>
2.1	Normalisation préalable . . . . .	6
2.2	Position d'un point par rapport à un élément . . . . .	6
2.3	Éléments coupés par un segment . . . . .	8
2.4	Retournement d'arêtes . . . . .	10
<b>3</b>	<b>Insertion de points dans un maillage existant</b>	<b>11</b>
3.1	La méthode de Voronoï . . . . .	11
3.2	Insertion locale et optimisation . . . . .	11
3.3	Insertion directe . . . . .	14
3.4	Insertion contrôlée par un critère . . . . .	16
<b>4</b>	<b>Quelques procédures d'accélération</b>	<b>17</b>
4.1	Grille de voisinage . . . . .	17
4.2	Transport des centres des disques circonscrits . . . . .	17
4.3	Mise à jour des voisins . . . . .	18
<b>5</b>	<b>Respect d'une frontière donnée</b>	<b>20</b>
5.1	Différentes définitions du problème . . . . .	20
5.2	Une solution naturelle . . . . .	21
5.3	Une solution par retournement d'arêtes . . . . .	22
5.4	Définition de l'intérieur du domaine . . . . .	24
<b>6</b>	<b>Génération des points internes</b>	<b>25</b>
6.1	Qualité attendue . . . . .	25
6.2	Méthode algébrique par éléments . . . . .	26
6.2.1	Découpe arithmétique d'une arête . . . . .	27
6.2.2	Découpe géométrique d'une arête . . . . .	28
6.2.3	Filtrage des points . . . . .	29
6.2.4	Insertion des points . . . . .	29
<b>7</b>	<b>Régularisation et optimisation</b>	<b>30</b>
7.1	Régularisation . . . . .	30
7.1.1	Barycentrage . . . . .	30
7.1.2	Bougé optimal . . . . .	31
7.2	Optimisation . . . . .	32
<b>8</b>	<b>Schéma du mailleur</b>	<b>32</b>
8.1	Quelques exemples . . . . .	32
8.2	Conclusion . . . . .	36
<b>9</b>	<b>Application à la dimension 3</b>	<b>39</b>



# 1 Introduction

L'ambition de ce papier est de présenter les différentes étapes intervenant dans le développement d'un mailleur 2D de type Delaunay en tentant de fixer "définitivement" la méthode. A ce titre, il remplace le rapport (épuisé) "Génération de maillages par une méthode de type Voronoï" (cf. [9]), en montrant comment appliquer les résultats décrits dans le rapport "Encore Delaunay" (cf. [3]), résultats relatifs à la méthode de Delaunay proprement dite qui constitue une part importante pour un tel mailleur. Nous espérons que le lecteur avisé prendra plaisir à constater l'évolution au niveau algorithmique entre ce papier et son prédécesseur et percevra le caractère "définitif" de la présente description.

Le problème que l'on veut résoudre est le suivant (on se place délibérément en dimension 2) : *Trouver un maillage acceptable d'un domaine quelconque à partir, essentiellement, de la donnée de son contour sous la forme d'un ensemble de segments.*

Le maillage obtenu sera naturellement le support d'un calcul qui pourra indiquer s'il est adapté ou s'il doit l'être par quelque méthode que ce soit (ce n'est pas l'objectif de ce travail). Il est donc important de débiter avec un bon maillage de départ sachant que cette requête de qualité ne peut être que basée sur des considérations géométriques puisque les seules informations supposées connues sont de nature géométrique. La forme et la taille des éléments devront donc être conformes vis-à-vis de ces données.

Avant de présenter la méthode, on va d'abord définir, en section 2, quelques outils qui seront constamment utilisés dans la suite. La section 3 décrit d'une part l'algorithme classique d'insertion d'un point dans un maillage donné et d'autre part une variante robuste de ce dernier. Dans la quatrième section on propose quelques procédures d'accélération pour l'implantation des routines décrites dans la section précédente. Dans la cinquième on montre comment on peut imposer une arête dans le maillage et ainsi définir un premier maillage du domaine considéré. La sixième section est consacrée à la génération des points internes. La septième est un bref résumé des procédures de régularisation et d'optimisation que l'on peut appliquer au maillage résultant. Dans la huitième section on décrit le schéma du mailleur et on conclut sur quelques résultats numériques. Enfin dans la dernière section, on propose les extensions possibles au cas de la dimension 3.

## 2 Quelques outils

On se place dans le cas de maillages (notés  $T$ ) composés de triangles seulement. On note par  $K$  les triangles. Un triangle est défini par la liste orientée, dans le sens direct, de ses sommets (pour donner un signe à la surface), ses arêtes sont également orientées (pour donner un sens aux normales), sa surface est positive. On associe à chaque triangle  $K$  l'ensemble de ses voisins (par ses arêtes)  $V_i(K)$ ,  $i = 1, 3$ ; le premier étant le triangle vu par le sommet 1 du triangle donné (avec la convention suivante:  $V_i(K) = 0$  signifie que l'arête  $i$  (opposée au sommet  $i$ ) de l'élément  $K$  est frontière). De plus, on notera par  $B(K)$  le disque ouvert circonscrit au triangle  $K$  (figure 1).



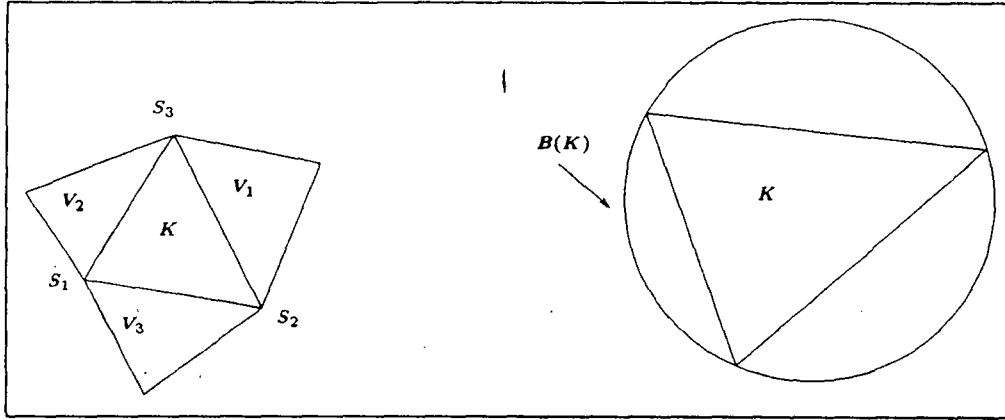


FIG. 1 - Définitions élémentaires.

## 2.1 Normalisation préalable

Pour éviter les erreurs d'arrondis, les données (i.e. les coordonnées des points fournis) sont normalisées entre 0 et 33554431 (i.e.  $2^{25} - 1$ ) puis converties en entiers.

Cette valeur représente la résolution ou pouvoir de séparation de la méthode. Elle est choisie de façon à permettre le calcul exact des surfaces (i.e., de longueurs au carré) et est donc fonction du nombre de bytes du mot de l'ordinateur employé. En effet les calculs seront faits via le type réel double précision défini sur 64 bits, soient 51 bits significatifs pour la mantisse; d'où la valeur  $25 = 51/2$ . Le calcul exact des surfaces est nécessaire pour une construction correcte de la triangulation; on montrera qu'il est suffisant pour la construction que nous proposons.

## 2.2 Position d'un point par rapport à un élément

Une opération usuelle consiste à *Trouver la situation d'un point P donné par rapport à un élément K donné.*

On considère  $K_j^* = \{F_j, P\}$ ,  $1 \leq j \leq 3$ , les 3 triangles virtuels construits en joignant l'arête  $F_j$  de  $K$  au point  $P$ . Si  $ABC$ , dans cet ordre, est le triangle  $K$ , alors les trois triangles associés sont successivement  $ABP$ ,  $APC$  et  $PBC$ . On définit par  $Det(K)$  le déterminant :

$$Det(K) = \begin{pmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{pmatrix},$$

où  $x_1, x_2, x_3$  sont les abscisses des sommets de  $K$  et  $y_1, y_2, y_3$  leurs ordonnées. Grâce à la normalisation précédente, le calcul de  $Det(K)$  est numériquement exact. Notons que cette valeur est le double de la surface de  $K$  (par la suite, de manière impropre, on fera la confusion entre  $Det(K)$  et la surface de  $K$ ).

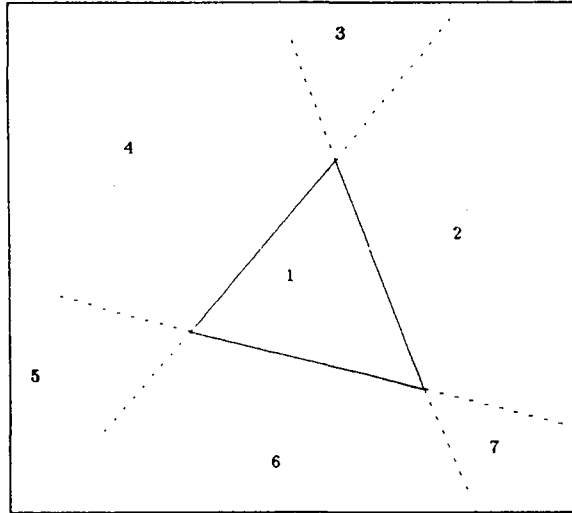


FIG. 2 - Position d'un point.

**Algorithme 1.**

- a) Si  $\forall j, 1 \leq j \leq 3, \text{Det}(K_{j_1}^*) > 0$  alors  $P \in K$ , Fin;
- b) Si  $\exists j_1, \dots, j_q, 1 \leq j_1 \neq \dots \neq j_q \leq 3$  tels que  $\text{Det}(K_{j_1}^*) = \dots = \text{Det}(K_{j_q}^*) = 0$  alors  $P$  est dans la région intersection des  $q$  droites déterminées par les arêtes  $F_{j_1} \dots F_{j_q}$ , Fin.
- c) Si  $\exists j_1, \dots, j_q, 1 \leq j_1 \neq \dots \neq j_q \leq 3$  tels que  $\text{Det}(K_{j_1}^*) < 0, \dots, \text{Det}(K_{j_q}^*) < 0$  alors  $P$  est dans la région intersection des  $q$  demi-plans définis par les arêtes  $F_{j_1} \dots F_{j_q}$  ne contenant pas les sommets opposés, Fin.  $\square$

**Proposition 1.**  $P$  est déterminé par rapport aux 7 régions définies par  $K$ .

**Démonstration.** Evidente par définition.  $\square$

Une application de cet algorithme permet de trouver à quel élément du maillage d'un domaine convexe appartient un point  $P$  interne à ce domaine. En effet à partir d'un triangle quelconque du maillage du domaine, on parcourt la triangulation par voisinage, pour aboutir à un triangle contenant  $P$ . Deux éléments consécutifs  $(K, V_j(K))$  dans ce parcours sont définis de telle manière que leur arête commune  $F_j$  sépare le premier élément  $K$  du point  $P$ . Ceci donne l'algorithme suivant :

**Algorithme 2.**

- a) Initialisation :  $K = K_0$  où  $K_0$  est un triangle quelconque du maillage;
- b) Si  $\forall j, 1 \leq j \leq 3, \text{Det}(K_j^*) > 0$  alors  $P \in K$  et  $\mathcal{C} = K$  Fin;
- c) Ou si  $A = \{j; \text{Det}(K_j^*) < 0\} \neq \emptyset$ , Considérer, d'une manière aléatoire, un élément  $j$  de  $A$ , faire  $K = V_j(K)$  et aller à b);
- d) Ou si  $\exists j_1, \dots, j_q$  tel que  $\text{Det}(K_{j_1}^*) = \dots = \text{Det}(K_{j_q}^*) = 0$  alors  $P \in (\bigcap_{k=1,q} (F_{j_k}))$  donc  $P \in \mathcal{C}$  où  $\mathcal{C}$  est l'ensemble des éléments du maillage incluant  $\bigcap F_{j_k}$  (qui est une arête si  $q = 1$  et un sommet si  $q = 2$ ), Fin.  $\square$

**Proposition 2. et définition.**  $\mathcal{C}$  est l'ensemble des éléments contenant  $P$  que l'on appelle base associée à  $P$ .

**Démonstration.** Par définition, il existe toujours un tel chemin reliant un triangle quelconque à un triangle contenant  $P$ ; à cause du caractère aléatoire de l'étape b), le chemin ainsi défini n'est pas cyclique et la convergence est garantie.  $\square$

Ici, en dimension 2, seules 3 situations sont possibles :

1.  $P \in K$ ,
2.  $P$  est sur une arête,
3.  $P$  est un sommet (en pratique on ne rencontrera pas ce cas, sauf pour les points multiples).

Donc la solution au problème (en excluant ce dernier cas) est :  $P \in K$  ou  $P \in \mathcal{C}$ , les 2 éléments contenant l'arête  $\bigcap F_{j_k}$ .

**Remarque.** Si dans b) les éléments de  $A$  sont ordonnés et si on considère systématiquement le premier, le chemin ainsi défini peut être cyclique et ne jamais atteindre un élément contenant  $P$ , comme le montre l'exemple suivant (figure 3).

A partir du triangle 1, les triangles 2, 3, 4, 5, 6 sont visités pour aboutir de nouveau au triangle 1. De même la situation est bloquée si on décide de ne pas parcourir les triangles déjà visités. Une situation différente (de l'algorithme) consiste à marquer les triangles visités et dans le cas où on arrive de nouveau dans un triangle déjà visité, on choisit un triangle quelconque non marqué pour réinitialiser le processus.

### 2.3 Eléments coupés par un segment

On se place dans le cas d'un domaine convexe dont on connaît un maillage et on cherche à *Trouver tous les éléments du maillage coupés par un segment donné*. En pratique et dans la suite, le but est de trouver tous les éléments coupés par un segment dont les extrémités  $A$  et  $B$  sont deux sommets du maillage. Pour cela, à partir d'un élément possédant  $A$  comme sommet et intersectant le segment  $AB$ , on retrouve par voisinage, les autres éléments qui intersectent  $AB$ .

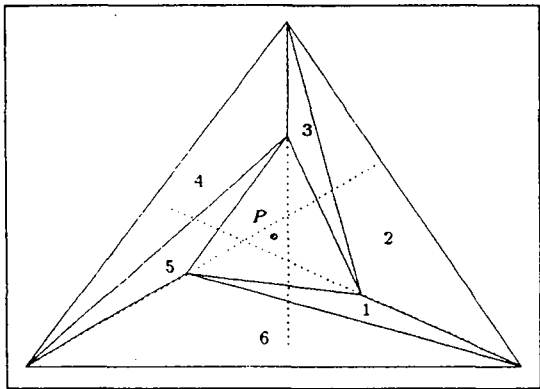


FIG. 3 - configuration cyclique

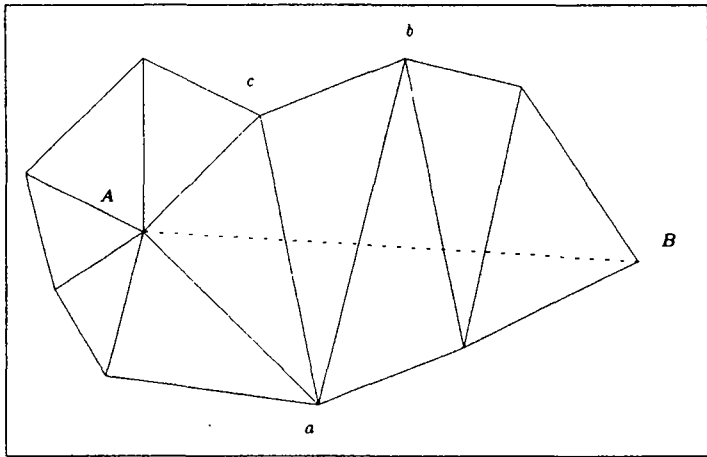


FIG. 4 - Triangles coupés par un segment.

**Algorithme 3.**

- a) Trouver un élément  $K$  ayant  $A$  comme sommet (si  $B$  est aussi sommet de  $K$ , l'ensemble cherché se réduit à cet élément  $K$ , Fin);
- b) Parmi les éléments ayant  $A$  comme sommet, trouver celui dont l'arête  $ac$  (cf. figure 4) opposée à  $A$  est coupée par  $AB$ , empiler cet élément;
- c) Prendre le voisin de  $K$  par l'arête  $ac$ , empiler cet élément;
- d) Soient  $abc$  les trois sommets de l'élément tête de pile, trouver parmi  $ab$  et  $bc$  (l'arête  $ac$ , coupée par  $AB$  est déjà connue) le couple tel que les puissances de  $a$  et  $b$  (respectivement  $b$  et  $c$ ) par rapport à  $AB$  soient de signe opposé et prendre le voisin par l'arête ainsi repérée, empiler cet élément;
- e) Tant que l'élément tête de pile n'a pas le point  $B$  comme sommet, aller en d), sinon Fin. □

**Proposition 3.** La pile ainsi constituée est l'ensemble cherché.

**Démonstration.** En effet l'intersection de deux segments de droite est soit vide, soit un point. □

Remarquons que le calcul des puissances utilisées ci-dessus est équivalent au calcul des quantités *Det* incriminées.

## 2.4 Retournement d'arêtes

Soient 2 triangles adjacents par une arête dont la réunion est un polygone convexe, on peut alors remailler ce polygone en créant l'arête formée par les 2 sommets opposés à l'arête initialement commune et en supprimant cette dernière (figure 5).

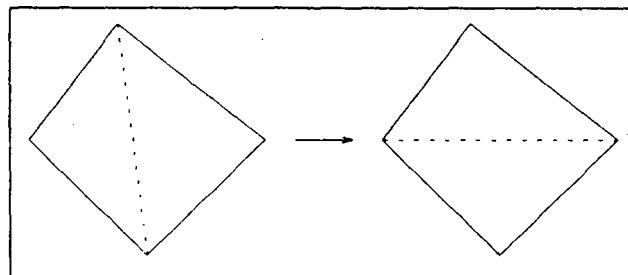


FIG. 5 - Retournement d'arête.

### 3 Insertion de points dans un maillage existant

Dans le cas d'un domaine convexe, on se donne un maillage  $T$  et un point  $P$  interne (dans le cas de plusieurs points, on appliquera la méthode à chacun). Le problème posé est de *Construire un nouveau maillage du domaine de telle sorte que  $P$  soit sommet d'éléments.*

#### 3.1 La méthode de Voronoï

On suppose ici que  $T$  est un maillage de Delaunay, i.e., le disque ouvert circonscrit,  $B(K)$ , à chacun de ses éléments  $K$  ne contient aucun point<sup>1</sup> (sommet de triangles du maillage) de  $T$ . La méthode de Voronoï consiste à construire un nouveau maillage possédant  $P$  comme sommet de triangles, tout en respectant la caractéristique ci-dessus; pour cela on tourne l'ensemble des triangles à disques non vide; cet ensemble forme un polygone étoilé par rapport à  $P$ . Il suffit alors de retriangler ce polygone en prenant en compte  $P$ ; chaque nouveau triangle est défini à partir de  $P$  et d'une arête frontière de ce polygone. La procédure peut se schématiser de la façon suivante:

##### Algorithme 4.

- a) Chercher les éléments  $K$  de  $T$  vérifiant  $P \in B(K)$  et les empiler dans  $S$ ;
- b) Trouver  $F_1 \dots F_p$  les arêtes externes de  $S$  c'est-à-dire les arêtes non communes à 2 éléments de  $S$ ;
- c) Construire le nouveau maillage comme  $T = (T - S) \cup \{F_j, P\}_j \quad 1 \leq j \leq p$ , Fin.  
□

**Proposition 4.**  $T$  est une triangulation de Delaunay incluant  $P$  comme sommet de triangles.

**Démonstration.** Voir [14] ou [15] (entre autres).

**Remarque.** L'étape a) utilise l'algorithme 2 décrit ci-dessus.

#### 3.2 Insertion locale et optimisation

Sans hypothèse sur la nature de  $T$ , le même problème peut être résolu en effectuant des retournement d'arêtes sur les arêtes frontières du polygone initialement constitué des éléments de la base, et en itérant le procédé si ce polygone est modifié. En effet un retournement d'arête permet de valider localement le critère de Delaunay et grâce au

<sup>1</sup>Il s'agit ici de l'une des caractérisations des maillages de Delaunay.

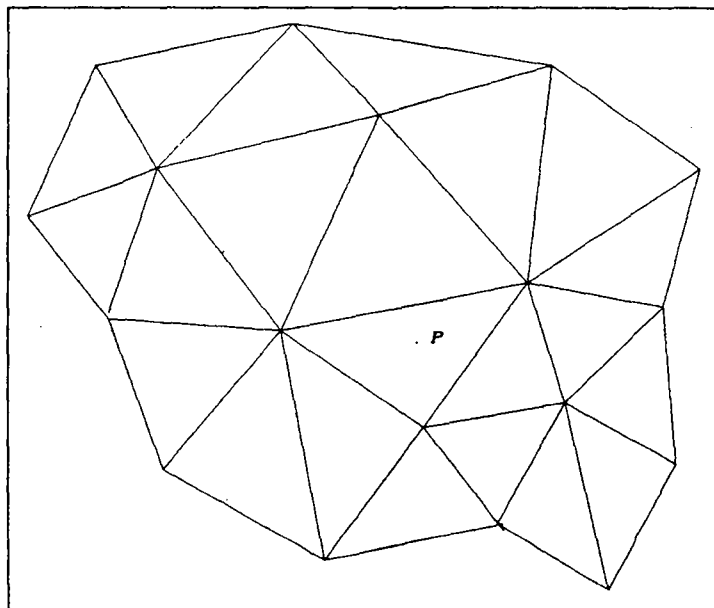


FIG. 6 - *Maillage initial et point à insérer.*

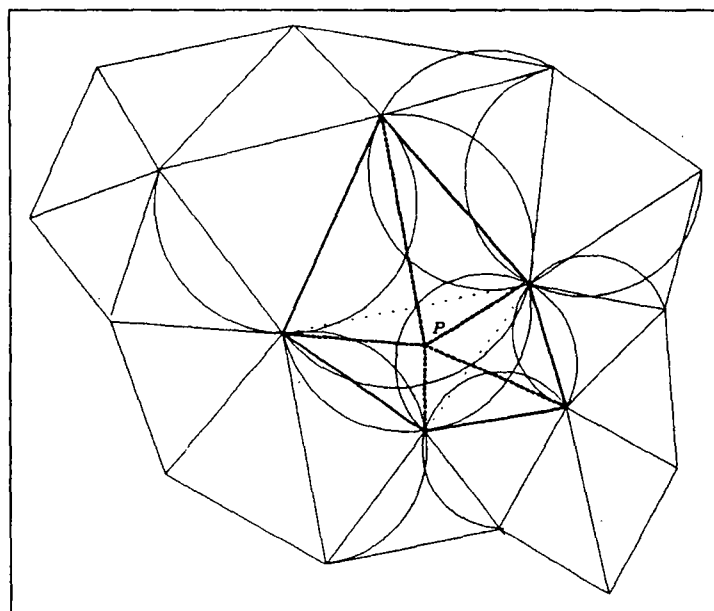


FIG. 7 - *Insertion d'un point (méthode 1).*

lemme général de Delaunay [6] le critère est aussi vérifié globalement.

#### Algorithme 5.

- a) Trouver la base associée à  $P$ ;
- b) Mettre le(les) élément(s) de la base dans une pile  $S$ ;
- c) Trouver  $F_1 \dots F_p$  les arêtes externes de  $S$  et joindre  $P$  à ces arêtes; on crée de la sorte 3 ou 4 éléments;
- d) Vider  $S$  et mettre ces éléments dans  $S$ ;
- e) Chercher récursivement parmi les voisins des éléments de  $S$  ceux dont le disque ouvert circonscrit contient  $P$  et retourner l'arête externe de  $S$  commune à ces éléments, mettre à jour  $S$  et tant que  $S$  est non vide aller en e), sinon Fin.  $\square$

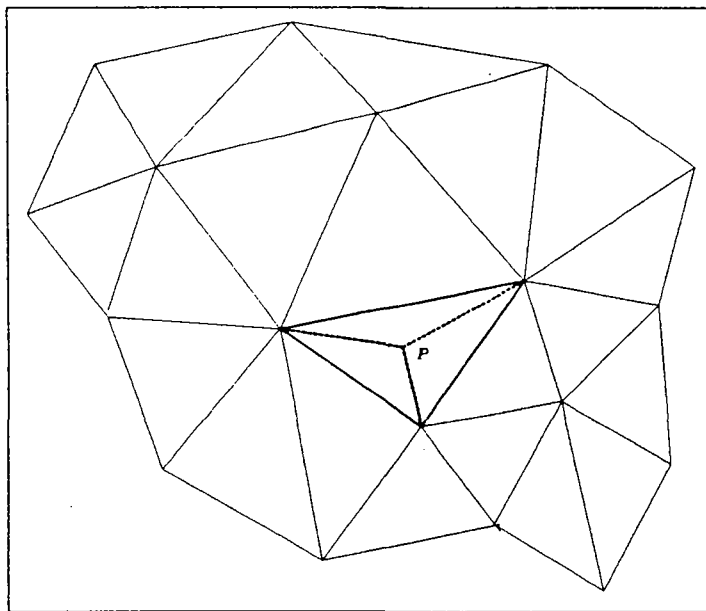


FIG. 8 - Insertion d'un point (méthode 2).

**Proposition 5.** Cet algorithme est constructif.

**Démonstration.** Evidente car la méthode est en elle-même constructive.

**Remarques.** L'étape a) utilise l'algorithme 2 décrit ci-dessus. L'étape e) utilise la procédure de retournement d'arêtes vue ci-dessus et s'interprète comme une optimisation locale. Le nouveau maillage est localement de Delaunay.



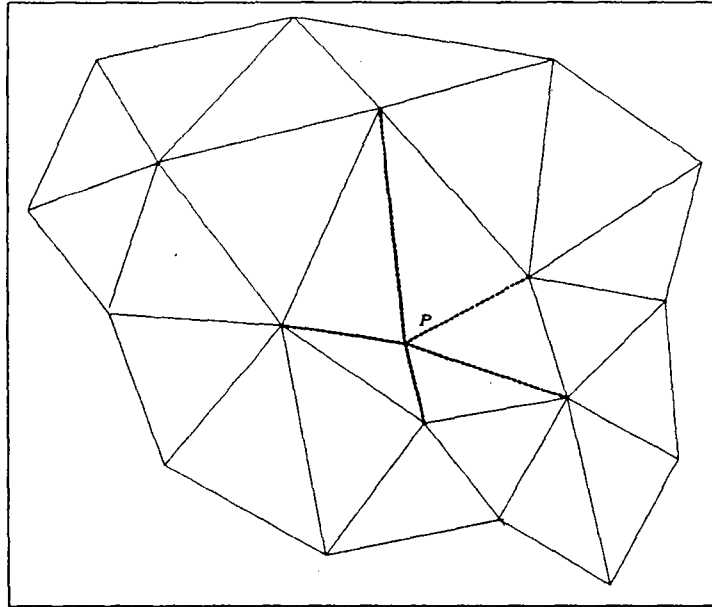


FIG. 9 - Optimisation (méthode 2).

### 3.3 Insertion directe

La méthode de Voronoï est numériquement peu fiable car savoir situer un point par rapport à un cercle (comparer  $P$  et  $B(K)$ ) ne conduit pas à une réponse claire dès lors que  $P$  est très voisin de  $B(K)$ . Une erreur minime (par exemple de 0.01 pour cent) sur le calcul correspondant entraîne généralement une réponse fausse (i.e. une erreur de 100 pour cent) à la question et la méthode produit un maillage généralement faux.

L'exemple suivant montre un cas de figure où le polygone constitué des triangles dont le disque ouvert circonscrit contient  $P$ , n'est pas étoilé par rapport à  $P$  et contient dans son intérieur un sommet (figure 10); dans ce cas le calcul numérique situe le point  $P$  dans l'intérieur des disques ouverts circonscrits aux triangles  $ACB$ ,  $AED$ ,  $AGE$ ,  $FGA$  et  $FEG$ .

Pour surmonter ce handicap, on apporte une correction au polygone constitué par l'ensemble des triangles à disques non vides; on garantit ainsi l'étoilement de ce polygone rapport à  $P$  et la non existence d'autre point, excepté  $P$ , à l'intérieur de ce polygone.

Ainsi pour l'exemple ci-dessus, il suffit de retirer les triangles  $ACB$  et  $FEG$  de l'ensemble des triangles à disques non vides pour retrouver les propriétés attendues; en effet dans ce cas le polygone cherché est alors celui défini par  $AFGED$ .

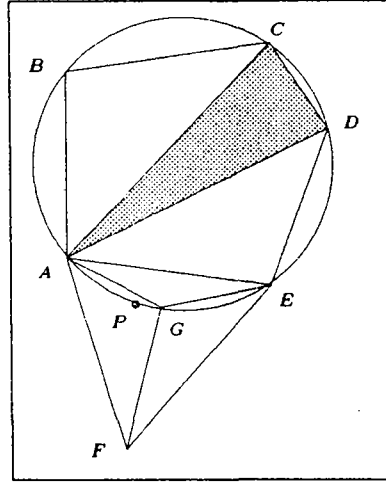


FIG. 10 - configuration ambiguë.

**Algorithme 6.**

- a) Trouver la base associée à  $P$ ;
- b) Mettre le(les) élément(s) de la base dans une pile  $S$ ;
- c) Compléter récursivement  $S$  en y ajoutant tout élément  $K$  de  $T$  tel que  $P \in B(K)$  aux erreurs de calcul près;
- d) Trouver  $F_1 \dots F_p$  les arêtes externes de  $S$  c'est-à-dire les arêtes non communes à 2 éléments de  $S$ ; Définir les  $p$  triangles virtuels  $K_j^* = (F_j, P), 1 \leq j \leq p$ ; Définir  $V_F$  l'ensemble des sommets des arêtes  $F_j$ ; Définir  $V_S$  l'ensemble des sommets des éléments de  $S$ ;
- e) Initialiser  $S' = S$ ;
- f) Pour  $j$  de 1 à  $p$ :
  - Si  $\text{Det}(K_j^*) > 0$ , ne rien faire;
  - Si  $\text{Det}(K_j^*) \leq 0$ ,  $S' = S' - K$  où  $K$  est le triangle de  $S'$  d'arête  $F_j$ ; par définition,  $K$  n'est pas contenu dans la base;
- g) Si  $S'$  n'a pas été modifié par l'étape f), aller en h); sinon mettre à jour l'ensemble des  $F_j$ ,  $V_F$  et  $V_{S'}$ , puis aller en f);
- h) Dénombrer  $V_F$  et  $V_{S'}$ ;  
 Si ces 2 ensembles ont le même cardinal faire  $S = S'$  et aller en i);  
 Sinon faire  $S' = S' - K$  et aller en f) ( $K$  est un triangle quelconque de  $T$  non contenu dans la base, possédant un sommet  $\neq P$ , contenu dans  $V_{S'}$  sans être dans  $V_F$ );
- i) Construire le nouveau maillage comme  $T = (T - S) \cup \{F_j, P\}_j, 1 \leq j \leq p$ , Fin.  
 $\square$

**Proposition 6.** Cet algorithme est constructif.

**Démonstration.** Par définition, la méthode est constructive et converge; en effet à chaque "correction" un élément non contenu dans la base, est enlevé de  $S$  et ainsi au pire  $S$  coïncide avec la base.

**Remarques.** L'étape a) utilise l'algorithme 2 décrit ci-dessus. L'étape f) ne repose que sur le calcul des  $Det$  correspondants. L'étape h) revient à effectuer un dénombrement.

Le nouveau maillage est correct (seuls des calculs entiers (donc exacts) ont été utilisés) et est presque de Delaunay. On en déduit que les seuls calculs exacts des surfaces sont suffisants pour construire le maillage, prenant en compte le point inséré.

### 3.4 Insertion contrôlée par un critère

Une écriture plus abstraite des schémas présentés est la suivante :

On se donne :

1.  $T$  un maillage d'un domaine contenant le point  $P$ ,
2.  $C_r$  un critère (ou un ensemble de critères),
3.  $C_o$  une contrainte (ou un ensemble de contraintes).

On isole dans  $T$  l'ensemble  $S$  des éléments tels que :

(H1)  $S$  vérifie le critère  $C_r$  pour le point  $P$

(H2)  $S$  satisfait la contrainte  $C_o$ .

Sous la **seule condition** que l'on sache remailler  $S$  en prenant en compte le point  $P$  (soit  $S'$  l'ensemble des éléments de ce remaillage local) on obtient alors le nouveau maillage  $T$  par :

$$T = T - S \cup S'.$$

Un choix adéquat de  $C_r$  permet d'assurer la validité du processus d'insertion du point  $P$ ; le choix de  $C_o$  permet d'ajouter des propriétés supplémentaires au maillage ainsi construit.

Dans le cas standard, on a le choix :

$$C_r = \{S \text{ vérifie } C_r \text{ pour le point } P\}$$

$$\Leftrightarrow P \text{ est dans le disque ouvert circonscrit de tout élément de } S\}$$

et on n'impose pas de contrainte.

Le cas 2 revient au même. Le cas 3 est en fait une interprétation constructive du critère qui s'écrit simplement comme :

$$C_r = \{S \text{ vérifie } C_r \text{ pour } P \Leftrightarrow \text{ses arêtes externes sont visibles par } P\}$$

ou, autrement dit :

$$C_r = \{S \text{ vérifie } C_r \text{ pour le point } P \Leftrightarrow P \text{ étoile l'ensemble } S\},$$

toujours sans imposer de contrainte.

Il est donc clair qu'un choix judicieux du couple  $\{C_r, C_o\}$  va permettre d'obtenir une triangulation possédant déjà des propriétés intéressantes, en particulier de façon à respecter une frontière donnée ou pour contrôler la qualité des éléments générés.

## 4 Quelques procédures d'accélération

Dans cette section on se propose d'analyser les différentes routines de l'algorithme 6 et d'apporter des méthodes rapides pour les rendre efficaces.

### 4.1 Grille de voisinage

Pour trouver un élément de la base, on parcourt la triangulation par voisinage à partir d'un triangle quelconque  $K_0$ . En choisissant de dernier assez proche de  $P$ , on réduit la longueur du chemin reliant  $K_0$  à  $P$ , donc le nombre de triangles dans le parcours; il suffit alors que l'un des sommets de  $K_0$  soit assez voisin de  $P$ . Ainsi pour chaque point, il faut définir ses points voisins. Pour cela, nous introduisons une grille de voisinage.

La grille est définie par un ensemble de rectangles uniformes, de côtés parallèles aux axes de coordonnées, que l'on appellera cases. La longueur de chaque côté est déterminée par la densité du nuage initial des points à trianguler. Dans le cas où la variation de la densité des points n'est pas très importante, on peut imposer le nombre maximal de points par case.

A chaque case on associe les points déjà insérés dans la triangulation et à chaque point déjà inséré, l'un des triangles de la triangulation possédant ce dernier comme sommet. Une case  $c$  est identifiée par un vecteur  $(c_{i_x}, c_{i_y})$  d'entiers. L'ensemble  $E_k(c)$  des cases voisins de degré  $k$  de  $c$  est défini par

$$E_k(c) = \{c' \text{ case}; \exists \alpha, 1 \leq \alpha \leq 2; |c'_{i_\alpha} - c_{i_\alpha}| \leq k\}.$$

Pour rechercher un point  $Q$  voisin de  $P$ , il suffit alors de trouver la case contenant  $P$ , puis de considérer l'un des points contenu dans cette case si elle est non vide et sinon, explorer les cases voisines dans l'ordre croissant du degré. Le triangle  $K_0$  est alors celui associé à  $Q$ .

Notons que l'utilisation de la grille n'est efficace que si le nombre de points est assez grand; dans ce cas la recherche via la grille ne sera déclanchée qu'à partir d'un seuil donné qui représente le nombre de points déjà insérés.

Il existe d'autres méthodes pour la recherche d'un élément de la base, qui considèrent des structures plus complexes que la grille. Par exemple, en conservant la hiérarchie de la construction du maillage, Boissonnat [2] montre que cette recherche s'effectue en un temps optimal si les points sont insérés dans un ordre aléatoire.

### 4.2 Transport des centres des disques circonscrits

Pour répondre à la requête  $P \in B(K)$  pour un triangle quelconque  $K$ , on est amené à associer à chaque triangle, le centre et le (carré du) rayon de son disque circonscrit; en

effet :

$$P \in B(K) \iff \|\vec{OP}\|^2 - r^2 < 0,$$

où  $O$  (resp.  $r$ ) est le centre (resp. le rayon) du disque circonscrit à  $K$ .

A chaque nouveau triangle  $K_j^* = \{F_j, P\}$ , on peut associer un ancien triangle  $K_j$ , contenu dans  $S$ , (vérifiant en particulier  $P \in B(K)$ ) tel que  $K_j^* = \{F_j, Q\}$ , où  $Q$  est le sommet de  $K_j$  non contenu dans  $F_j$ . Le centre et le rayon du disque circonscrit à  $K_j$  étant connus, on va montrer que l'on peut évaluer le centre et le rayon du disque circonscrit à  $K_j^*$ , à partir de  $K_j$ , par transport.

Soient  $n$  un vecteur normale à la droite support de  $F_j$  orienté vers  $Q$ ,  $O$  et  $r$  (resp.  $O^*$  et  $r^*$ ) le centre et le rayon du disque circonscrit à  $K_j$  (resp.  $K_j^*$ ). Par définition,  $O^*$  appartient à la droite de vecteur directeur  $n$  passant par  $O$ . Ainsi il existe un réel  $t$  vérifiant :

$$(1) \quad O^* = O + t * n.$$

Soit  $A$  l'un des sommets de  $F_j$ , on a :  $\|\vec{O^*P}\| = \|\vec{O^*A}\|$ . Si  $M$  est le milieu du segment  $AP$ , alors  $O^*$  appartient à la droite de vecteur normale  $\vec{AP}$  passant par  $M$ , ou :

$$(2) \quad \vec{AP} \cdot \vec{MO^*} = 0.$$

On peut déterminer  $t$  à partir des équations (1) et (2); en effet,  $MO^* = MO + t * n$  et  $\vec{AP} \cdot (\vec{MO} + t * n) = 0$ ; on a  $\vec{AP} = \vec{OP} - \vec{OA}$ ,  $\vec{OM} = \frac{\vec{OP} + \vec{OA}}{2}$  et on choisit  $n$  tel que  $Det(K_j^*) = \vec{AP} \cdot n$ ; comme  $\|\vec{OA}\| = r$ , on en déduit :

$$t = \frac{\|\vec{OP}\|^2 - r^2}{2 * Det(K_j^*)};$$

en reportant  $t$  dans (1) on obtient  $O^*$  et on a :  $r^{*2} = \|\vec{O^*P}\|^2$ .

Nous remarquons que les quantités  $\|\vec{OP}\|^2 - r^2$ , associée au critère de Delaunay, et  $Det(K_j^*)$ , associée à un critère de visibilité, sont déjà calculées.

Cette formule permet d'éviter le calcul de  $O^*$  via la résolution du système linéaire défini par les équations  $\|\vec{O^*A}\| = \|\vec{O^*B}\|$  et  $\|\vec{O^*A}\| = \|\vec{O^*C}\|$ , où  $B$  et  $C$  sont les autres sommets de  $K_j^*$ .

### 4.3 Mise à jour des voisins

A chaque triangle, sont associés ses trois voisins; ainsi pour chaque nouveau triangle, il faut définir les relations de voisinage. Remarquons que si  $K_j^* = \{F_j, P\}$  et  $K_i^* = \{F_i, P\}$  sont voisins alors les arêtes  $F_j$  et  $F_i$  ont un sommet en commun. Ainsi si on arrive à imposer  $i = j + 1$  ou en d'autre terme si les arêtes  $F_i$  et  $F_j$  sont empilées l'une après l'autre, alors pour un triangle, on peut établir automatiquement ses voisins; à savoir  $K_j^*$  est voisin à la fois de  $K_{j-1}^*$  et de  $K_{j+1}^*$ .

Pour cela il suffit d'empiler les éléments de  $S$ , (vérifiant  $P \in B(K)$ ) tout en conservant un sommet d'un élément de la base, jusqu'à l'obtention de l'arête  $F_1$  qui contient alors ce sommet, puis empiler les éléments, tout en conservant l'autre sommet de  $F_1$  jusqu'à l'obtention d'une nouvelle arête  $F_2$  et itérer ce procédé à partir de  $F_2$ . En appliquant le procédé à la dernière arête, on retrouve  $F_1$  et le processus est terminé.

Pour illustrer le procédé on considère un exemple. Soit la configuration des triangles de la figure 11, où  $ABC$  est le triangle de la base associée au point  $P$  et les disques ouverts circonscrits aux autres triangles contiennent  $P$ .

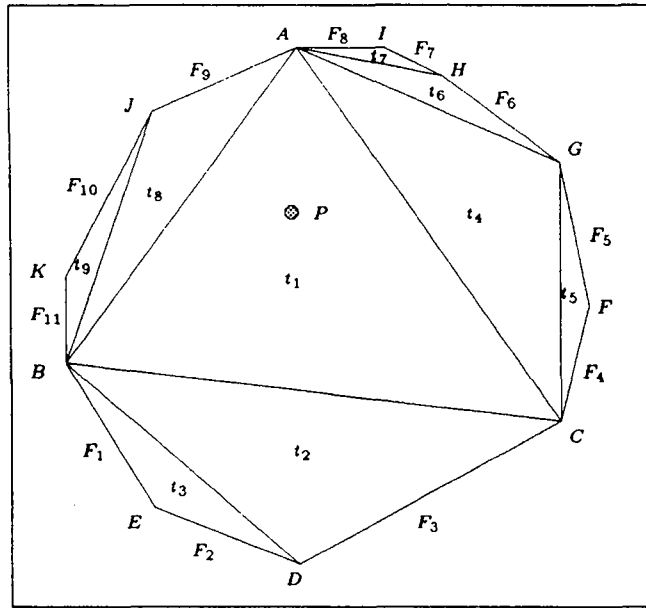


FIG. 11 - *L'empilement des éléments.*

Le problème est : comment empiler les triangles à partir de  $ABC$  pour que les arêtes frontières du polygone constitué par ces triangles arrivent dans un ordre cyclique.

Le triangle  $ABC$  est déjà empilé en tant qu'élément de la base; il constitue le premier élément  $t_1$  de la pile. Pour empiler les autres éléments, on applique au couple  $(t_1, A)$ , l'algorithme récursif suivant, qui considère le couple  $(t, X)$  constitué d'un triangle et d'un de ses sommets comme entrée :

- Soit  $t_X = V_X(t)$ , où  $V_X(t)$  est le voisin opposé au sommet  $X$  de  $t$ ; soit  $X'$  le sommet de  $t_X$ , successeur (dans le sens direct) du sommet opposé au sommet  $X$  de  $t$ ; empiler  $t_X$  et appliquer l'algorithme au couple  $(t_X, X')$ ;
- Soit  $Y$  le successeur de  $X$  dans  $t$ ;  $t_Y = V_Y(t)$ , soit  $Y'$  le sommet de  $t_Y$ , successeur du sommet opposé au sommet  $Y$  de  $t$ ; empiler  $t_Y$  et appliquer l'algorithme au couple  $(t_Y, Y')$ ;
- Soit  $Z$  le successeur de  $Y$  dans  $t$ ;  $t_Z = V_Z(t)$ , soit  $Z'$  le sommet de  $t_Z$ , successeur du sommet opposé au sommet  $Z$  de  $t$ ; empiler  $t_Z$  et appliquer l'algorithme au couple  $(t_Z, Z')$ .

En appliquant la procédure au couple  $(t_1, A)$  on obtient successivement  $(t_2, C)$ ,  $(t_3, D)$ ,  $F_1$ ,  $(t_3, B)$ ,  $F_2$ ,  $(t_2, B)$ ,  $F_3$ ,  $(t_4, A)$ ,  $(t_5, G)$ ,  $F_4$ , etc.

En effet le couple  $(t_2, C)$  est défini à partir de  $(t_1, A)$ ; le sommet à conserver dans l'empilement étant le successeur de  $A$  dans  $t_1$ , c'est-à-dire  $B$  et ainsi  $t_2 = V_A(t_1)$  et on a  $C =$  le successeur de  $D$  (sommet opposé à  $A$ ) dans  $t_2$ ; de même à partir de  $(t_2, C)$ , on obtient  $(t_3, D)$  et on trouve l'arête  $F_1$  car  $P \notin B(V_D(t_3))$ ; L'empilement continue de telle manière qu'à chaque étape on conserve successivement les sommets  $E, D, C, F, G, H, I, A, J$  et  $K$  jusqu'à l'obtention d'une arête  $F_i$  définie par un triangle dont le disque ouvert circonscrit ne contient pas  $P$ .

**Remarque.** L'empilement ainsi défini, supporte toute correction éventuelle; en effet en retirant des éléments de l'ensemble des triangles  $(t_i)$ , l'ordre d'empilement des autres éléments reste inchangé.

## 5 Respect d'une frontière donnée

La section 3 a montré comment insérer un point dans un maillage (pour un domaine convexe), le problème posé initialement est *sensiblement* différent. Il s'agit de mailler un domaine, non nécessairement convexe, à partir de la donnée de sa frontière, celle-ci étant connue comme l'ensemble des segments la constituant.

Une première idée consiste à :

- Construire l'ensemble des points  $P$  ci-dessus comme l'ensemble des points extrémités des arêtes de la frontière donnée;
- Construire 4 points supplémentaires tels que tout point donné soit interne au quadrilatère formé par ces 4 points;
- Mailler ce quadrilatère avec 2 triangles (de Delaunay);
- Appliquer l'un quelconque des processus d'insertion de points pour construire, pas à pas, un maillage du quadrilatère composé de triangles dont les sommets sont, d'une part les 4 points ajoutés et d'autre part tous les points de la frontière.

Ce maillage *ne contient pas*, en général, dans la liste des arêtes de ses éléments les arêtes données. Par suite, on ne peut pas définir correctement le domaine donné (les notions d'intérieur et d'extérieur ne peuvent, en effet, être précisées que par rapport à la frontière fournie donc par rapport aux arêtes de celle-ci).

La définition du domaine nécessite donc la présence de la frontière donnée dans le maillage issu de l'insertion, un par un, des points de cette frontière. Il existe plusieurs façons de poser ce problème et, par suite, plusieurs réponses possibles.

### 5.1 Différentes définitions du problème

La première idée consiste à trouver la régularité que doivent satisfaire les données de telle sorte que le problème ne se pose pas c'est-à-dire que le maillage contenant les extrémités des arêtes données contienne ces arêtes. On trouvera dans [12] l'introduction

de la notion d'arêtes Delaunay admissibles qui implique le résultat. Pratiquement cette notion est délicate, voire impossible, à traduire, par suite, elle est d'un intérêt numérique faible.

Une deuxième idée consiste à recenser les arêtes manquantes et à les couper en deux par création du point milieu. Le nouvel ensemble de points ainsi formé sert alors de donnée au processus d'insertion de points avec le secret espoir que le maillage obtenu contienne toutes les arêtes données (arêtes initiales ou arêtes initiales coupées en deux). Intuitivement, une application récursive du procédé donne le résultat car elle revient à former numériquement un ensemble de données Delaunay admissible. On trouvera dans [17] la mise en application de cette méthode et dans [23] une extension astucieuse.

Une autre idée consiste à transformer localement le maillage initial pour régénérer les arêtes manquantes (et supprimer les arêtes gênantes). Plusieurs manières de procéder existent, parmi lesquelles nous choisissons les deux méthodes exposées ci-dessous.

## 5.2 Une solution naturelle

Soit  $T$  le maillage précédent construit en insérant les points extrémités des arêtes données. Soit  $AB$  une arête de donnée non présente dans  $T$ , on définit  $T_{AB}$  l'ensemble des triangles dont une arête est coupée par  $AB$ . Ainsi  $T = T' \cup T_{AB}$  où  $T'$  est l'ensemble des triangles non intéressés par  $AB$ .

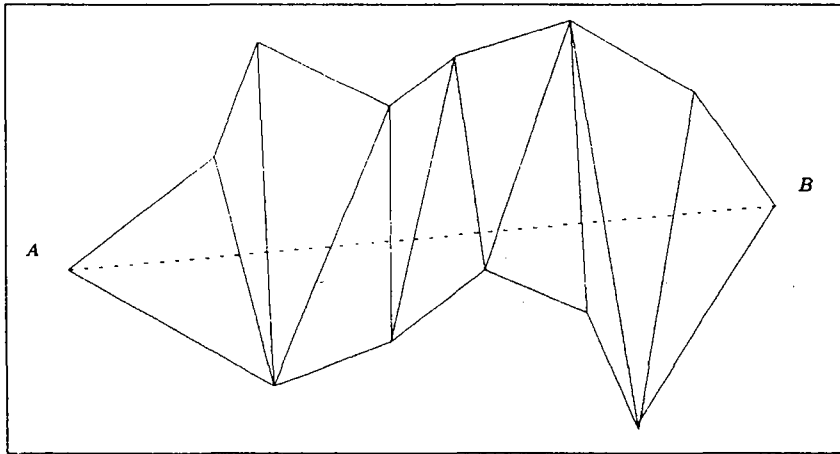


FIG. 12 - Ensemble de départ.

$T_{AB}$  définit un polygone (non convexe) dont on note les sommets par  $a_k, k = 1, n_1$  avec  $a_1 = A$  et  $a_{n_1} = B$  pour la partie située d'un côté de  $AB$  et  $b_k, k = 1, n_2$  avec  $b_1 = A$  et  $b_{n_2} = B$  pour la partie située dans le demi-plan délimité par  $AB$  ne contenant pas les  $a_k$ .

On note par  $Pb^0$  le problème à résoudre i.e., le remaillage de  $T_{AB}$ . Il est clair que, si on note  $Pb_1^0$  le problème du remaillage du polygone constitué par les  $a_k$  et  $Pb_2^0$  celui du remaillage du polygone formé par les  $b_k$ , le problème initial revient d'une part, à traiter indépendamment les deux sous-problèmes  $Pb_i^0$  dont la complexité est moindre, et d'autre



part, en fait à ne traiter que l'un de ces sous-problèmes. Considérons donc le problème  $Pb_1^0$ .

On trace  $AB$  et on cherche dans  $a_k$  le point le plus proche de  $AB$ , soit  $a_l$  ce point, on trace  $Aa_l$  et  $a_lB$  et on forme le triangle  $ABa_l$ . La solution du problème  $Pb_1^0$  est donc la suivante :

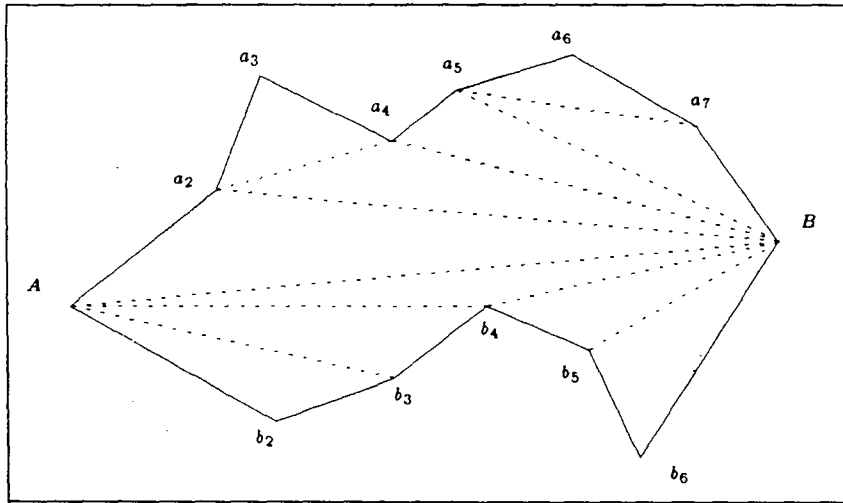


FIG. 13 - *Forçage naturel.*

- construire le triangle  $a_1, a_{n_1}, a_l$ ;
- définir le problème  $Pb_1^1$  qui est celui du maillage du polygone  $a_1, a_2, \dots, a_l$ ;
- définir le problème  $Pb_1^2$  qui est celui de maillage du polygone  $a_l, a_{l+1}, \dots, a_{n_1}$ .

Le problème est donc décomposé en sous-problèmes de complexité inférieure, appliquant récursivement le procédé, la solution est obtenue (un polygone réduit à trois points donne la solution et arrête la décomposition).

La description complète de cette méthode figure dans [11].

**Remarque.** Le polygone défini par  $T_{AB}$  peut éventuellement ne pas être simple comme le montre l'exemple de la figure 14.

Les sommets  $a_2$  et  $a_4$  de  $T_{AB}$  sont confondus.

### 5.3 Une solution par retournement d'arêtes

La méthode ci-dessus donne le résultat mais celui-ci est peu équilibré, par suite une autre solution est proposée, basée sur le procédé de retournement d'arête.

L'idée de base est la suivante : appliquer, d'une manière aléatoire, des retournements d'arêtes dans  $T_{AB}$ , pour créer des triangles qui n'intersectent pas  $AB$ .

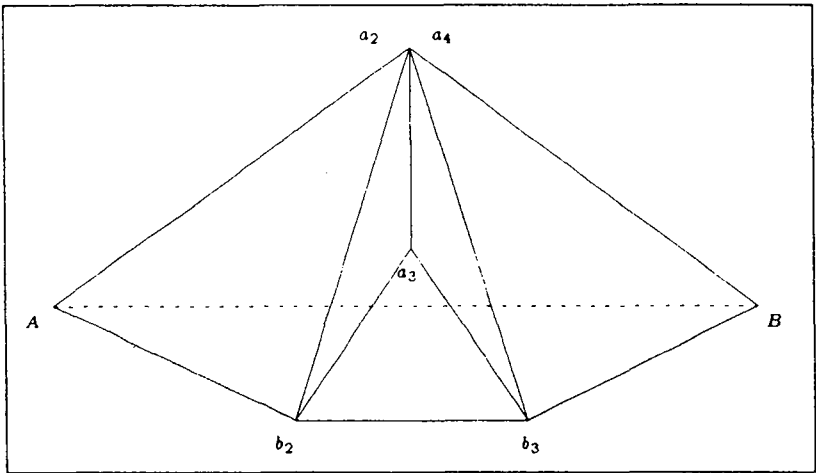


FIG. 14 -  $T_{AB}$  n'est pas simple.

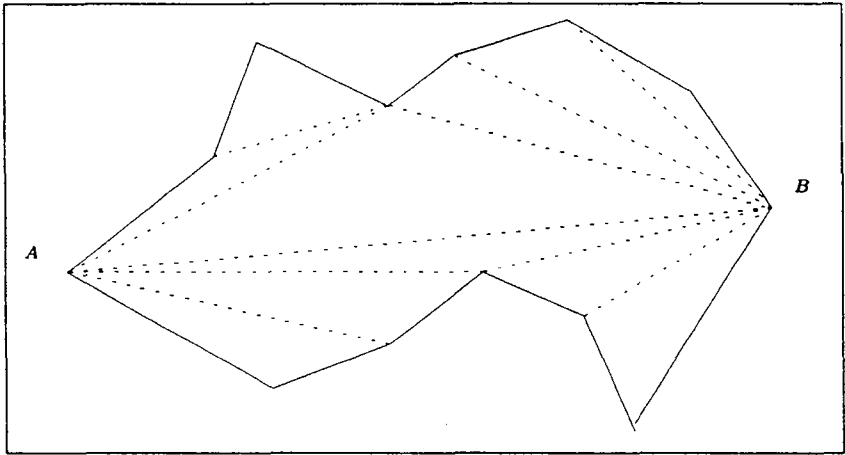


FIG. 15 - Forçage par retournements aléatoires.

La solution proposée est donc la suivante :

#### Algorithme 7.

- a) définir  $T_{AB}$ ;
- b) essayer de retourner une arête interne quelconque de  $T_{AB}$ ;
  - b1) le retournement marche, aller en c);
  - b2) le retournement ne marche pas, aller en b);
- c) mettre à jour  $T_{AB}$  (c'est-à-dire si un triangle  $K$ , non intersectant  $AB$ , est formé, faire  $T_{AB} = T_{AB} - \{K\}$ );
- d) si  $Card(T_{AB}) = 2$  et l'un des éléments de  $T_{AB}$  contient le segment  $AB$  alors Fin, sinon, aller en b) □

**Proposition 7.** Le polygone  $T_{AB}$  initial est remaillé et ce maillage contient  $AB$  comme arête de triangles.

**Démonstration.** On utilise un théorème, démontré par exemple dans [5], qui montre que l'on peut passer d'une triangulation quelconque  $T_1$  à une autre triangulation quelconque  $T_2$  d'un polygone, en effectuant des bascules d'arêtes. En fait on passe de  $T_1$  à une triangulation de Delaunay puis on effectue des retournements à l'envers pour passer à  $T_2$ , en constatant que le processus de retournement d'arêtes est réversible.

**Remarque.** On peut appliquer des optimisations locales aux arêtes internes, excepté  $AB$ , du nouveau maillage de  $T_{AB}$ .

## 5.4 Définition de l'intérieur du domaine

Comme  $T = T' \cup T_{AB}$  et que  $T_{AB}$  a été remaillé de telle sorte que  $AB$  existe,  $AB$  existe dans  $T$ . Sous la seule hypothèse que la frontière du domaine n'est pas croisée (ce qui est trivialement vérifié), on peut appliquer le processus à toutes les arêtes manquantes de façon à obtenir un maillage  $T$  contenant exactement toutes les arêtes données. Ce maillage permet de définir l'intérieur du domaine en construisant le graphe des composantes connexes

du domaine.

### Algorithme 8.

- a) Prendre un élément quelconque de  $T$ , le marquer par un entier  $n$ ;
- b) Examiner ses voisins :
  - Tant que l'arête commune n'est pas une arête frontière et que le voisin n'est pas déjà marqué, le marquer avec le nombre  $n$  et empiler le triangle si ses 3 arêtes ont été examinées;
- c) Tant que la pile d'indice  $n$  peut évoluer, aller en b);
- d) Sinon, tant qu'il reste des triangles non marqués, prendre un triangle non marqué, faire  $n = n + 1$ , le marquer avec  $n$  et aller en b). □

Ensuite, il suffira de retenir tous les triangles marqués par  $n$  pour avoir tous les éléments appartenant à cette composante du domaine (ceci permettra d'éliminer l'extérieur du domaine et sera effectué plus tard).

## 6 Génération des points internes

A ce stade, on dispose d'un maillage  $T$  dont les éléments sont marqués. Ce maillage ne contient comme sommets que les points de la frontière donc est généralement impropre aux calculs. Cette phase va donc consister à créer des points internes dans les éléments marqués et à les insérer dans  $T$ .

Pour procéder à cette phase, il faut répondre aux deux questions suivantes :

- Faut-il créer un (des) point(s) dans  $T$ ?
- Où mettre ce (ces) points?

La réponse à ces questions dépend étroitement de ce que l'on veut faire du maillage final en terme de calculs. Il est clair que la nature de celui-ci (mécanique du solide, simulation thermique, mécanique des fluides, ...) et que la nature des données physiques (caractéristiques des matériaux, vitesse, pression, nature des efforts, températures imposées, etc.) influent sur la nature attendue du maillage.

### 6.1 Qualité attendue

Que peut-t-on faire? Quel est le but visé et quels sont les paramètres sur lesquels on peut influencer? Le but visé est d'obtenir un maillage adapté aux calculs. Les paramètres sur lesquels on peut agir sont :

- La taille des triangles;

- La forme des triangles.

La seule donnée étant la description discrète de la frontière du domaine (et éventuellement la donnée de quelques points internes), on ne peut utiliser, pour créer  $T$ , que des informations relatives à cette donnée. Ainsi le but visé ne peut être *raisonnablement* que d'obtenir un maillage isotrope reflétant les propriétés des données.  $T$  sera donc adapté aux données et non, a priori au problème physique à résoudre. On ne peut pas faire mieux, sans choix discutables, en l'absence d'informations supplémentaires. Par suite, la seule ambition de la méthode proposée est de construire un maillage  $T$  raisonnable qui permettra d'effectuer un premier calcul du problème posé; le résultat de celui-ci et l'analyse des champs de solutions permettra alors d'adapter  $T$  au problème (voir par exemple [7] sur les méthodes d'adaptation et [20] pour le cas anisotrope ou directionnel).

## 6.2 Méthode algébrique par éléments

A chaque point donné, on associe un  $h$  local qui est la distance souhaitable entre ce point et ses voisins. Si le point est un point interne spécifié,  $h$  est fourni par l'utilisateur; si le point est un point de la frontière,  $h$  est la moyenne géométrique entre les longueurs des arêtes ayant ce point comme extrémité. S'il y a une variation importante entre les  $h$  de deux points consécutifs du contour, on peut alors considérer différentes méthodes de lissage de  $h$ .

Il existent plusieurs procédés de génération de points internes, par exemple celles définis dans [13][14]; Dans la plupart, le choix est de générer les points à l'intérieur des éléments suivant un critère de forme dépendant de  $h$ . On propose de générer les points internes sur les arêtes internes et dans certains cas à l'intérieur des triangles internes aux domaines à mailler. Ces points sont définis suivant des critères géométriques et des critères de proximités dépendant de  $h$ . On a le processus suivant :

### Algorithme 9.

- Faire, pour tout les éléments  $K$  de  $T$  :
  - Examiner les 3 arêtes de  $K$  (si celles-ci n'ont pas été déjà examinées et si elles ne sont pas des arêtes frontières), en comparant leurs longueurs  $l_i$  avec les  $h$  de leurs extrémités (soient  $h_{i1}$  et  $h_{i2}$  les  $h$  des extrémités de l'arête  $i$ );
  - Si  $l_i$  est plus grand que  $h_{i1} + h_{i2}$  calculer  $n_i$  le nombre de points à créer sur l'arête de telle sorte que ces points suivent une répartition donnée (voir ci-dessous) et générer ces  $n_i$  points;
  - Si pour une arête,  $n_i = 0$  et  $l_i > \gamma \max(h_{i1}, h_{i2})$ , alors générer un point au barycentre (pondéré par les  $1/h_i$ ) des sommets de  $K$ ;  $\gamma$  est un coefficient qui diminue l'effet correspondant au choc des  $h$  (sa valeur optimale s'avère être 0.65).
  - Prendre tous les points ainsi créés et empiler ceux qui ne sont pas trop proches (au sens de  $h$ ) d'un point existant ou déjà empilé.
- Fin. □

Dans la suite de cette section, nous allons donner quelques indications relatives à la génération des points sur les arêtes.

#### 6.2.1 Découpe arithmétique d'une arête

La découpe est déclenchée dans ce cas, si pratiquement il n'y a pas de variation de  $h$  entre les extrémités de l'arête. Soit  $F_k$  l'arête examinée et  $d_k$  sa longueur, on note par  $P_k(0)$  et  $P_k(n+1)$  ses 2 extrémités et par  $h(0)$  and  $h(n+1)$  les  $h$  correspondants, on définit la suite :

$$\alpha_i = d(P_k(i), P_k(i+1)) \text{ pour } i = 0, \dots, n$$

avec

$$\alpha_0 = h(0) + r \quad \text{et} \quad \alpha_n = h(n+1) - r$$

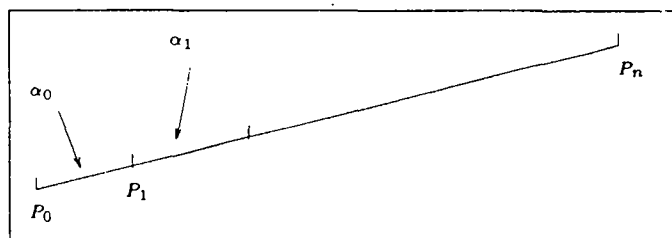


FIG. 16 - Découpe d'une arête.

Le problème est de calculer  $n$  et  $r$  donc les  $\alpha_i$  tels que la suite des distances ainsi définie soit arithmétique, c'est-à-dire que :

$$\alpha_{i+1} = \alpha_i + r \quad 0 \leq i \leq n$$

où  $r$  est la raison cherchée. Le système à résoudre est :

$$\begin{cases} \sum_{i=0}^n \alpha_i &= d_k \\ (n+2)r &= \Delta H = |h(n+1) - h(0)| \end{cases} \quad (1)$$

Ce système traduit que  $d_k$  est la somme des longueurs des sous-segments que l'on cherche à créer pour découper l'arête  $F_k$  et que la variation de  $h(0)$  à  $h(n+1)$  est équi-répartie entre tous ces segments.

On en déduit :

$$\begin{cases} (n+1)(h(0) + \frac{(n+2)}{2}r) &= d_k \\ (n+2)r &= \Delta H \end{cases} \quad (2)$$

Pour trouver  $n$ , on exprime  $r$  en fonction de  $n$  et on obtient :

$$n = \frac{2d_k}{h(0) + h(n+1)} - 1$$

Toutefois,  $n$  doit être entier ce que ne donne pas ce calcul, par suite, on choisit pour  $n$  la valeur entière la plus proche de la valeur calculée.  $n$  étant connue, on calcule  $r$  :

$$r = \frac{|h(n+1) - h(0)|}{n+2}$$

Le couple  $(n, r)$  ainsi obtenu correspond à une distance  $d'_k = \frac{(n+1)(h(0) + h(n+1))}{2}$ .

Le facteur de correction  $d_k/d'_k$  permet la remise à l'échelle des distances  $\alpha_i$ .

La valeur de  $h$  en chaque point  $P_k(i)$ ,  $1 \leq i \leq n$ , est définie par :

$$h(P_k(i)) = \frac{d(P_k(i), P_k(0))h(P_k(0)) + d(P_k(i), P_k(n+1))h(P_k(n+1))}{d_k}$$

### 6.2.2 Découpe géométrique d'une arête

La même analyse est appliquée dans le cas où une progression géométrique est souhaitée. Cette dernière est appliquée s'il existe une variation de  $h$  entre les extrémités de l'arête; en effet dans ce cas, la variation des  $h$  pour les points ainsi définis est plus lisse par rapport à une découpe arithmétique. On a :

$$\alpha_{i+1} = r\alpha_i$$

avec

$$\alpha_0 = r h(0) \quad \text{et} \quad \alpha_n = r^{-1} h(n+1)$$

Dans ce cas, le système à résoudre est :

$$\begin{cases} \sum_{i=0}^n \alpha_i &= d_k \\ h(n+1) &= r^{n+2} h(0) \end{cases} \quad (3)$$

qui conduit à :

$$\begin{cases} h(0) r \frac{1-r^{n+1}}{1-r} &= d_k \\ r^{n+2} &= \frac{h(n+1)}{h(0)}; \end{cases} \quad (4)$$

$r$  est calculé, puis  $n$  est obtenu. On obtient successivement :

$$r = \frac{d_k + h(n+1)}{d_k + h(0)},$$

$$n = \frac{\ln \frac{h(n+1)}{h(0)}}{\ln(r)} - 2.$$

De même  $n$  est remplacé par l'entier le plus proche et le facteur de correction est évalué. Les différents  $\alpha_i$  sont alors connus et les points intermédiaires, ainsi que leurs  $h$ , sont parfaitement déterminés.

### 6.2.3 Filtrage des points

Appliquant l'opération à tous les triangles marqués de  $T$ , on dispose, à ce stade, d'une série de points placés sur les arêtes de  $T$  et éventuellement à l'intérieur de ses éléments.

Le résultat est un ensemble de points bien situés localement (dans chaque triangle) mais, à cause de la localité du procédé, il est en général possible d'avoir généré des points voisins qui violent mutuellement les  $h$  correspondants. Pour éviter ce problème, on examine maintenant tous les points afin de supprimer ceux qui ne sont pas nécessaires.

Pour ce faire, on utilise la grille de voisinage et on code dans ses cases les points présents; chaque case est analysée : si elle contient plusieurs points, ceux-ci sont comparés et supprimés le cas échéant (en fonction de leur  $h$  à un facteur  $\gamma$  près); sinon on regarde les cases voisines dans l'ordre croissant du degré tant que les points présents sont trop proche au regard des  $h$  des points examinés.

### 6.2.4 Insertion des points

Les points retenus sont insérés, d'une manière aléatoire, dans  $T$  en utilisant les algorithmes décrits dans la section 3 et les arêtes de la nouvelle triangulation sont à nouveau



examinées; Si l'ensemble des points générés est non vide, on itère le processus. On obtient :

#### Algorithme 10.

- Faire, d'une manière aléatoire, pour tous les points de la pile :
  - Insérer le point dans  $T$ ;
- Réexaminer les arêtes de  $T$ .
- Si la pile correspondante aux nouveaux points est non vide itérer, sinon Fin.  $\square$

**Proposition 8.** L'algorithme converge.

**Démonstration.** En effet les longueurs des arêtes diminuent et ne peuvent pas dépasser les seuils fixés par les  $h$  initiaux.

**Remarque.** Le caractère aléatoire pour l'insertion des points réduit en moyenne, à chaque étape d'insertion, le nombre de triangles dont le disque ouvert contient le point inséré.

## 7 Régularisation et optimisation

A ce stade, on apporte des modifications locales au maillage pour améliorer une mesure ou autre de la qualité des éléments obtenus (il faut noter que notre souci pour la génération des points internes était de surmonter l'étape de régularisation et optimisation; néanmoins dans des cas rares ou pour la beauté du geste ces dernières procédures sont appliquées).

### 7.1 Régularisation

La régularisation consiste à bouger les sommets du maillage d'une part pour augmenter la qualité des éléments et d'autre part pour rendre plus lisse la variation des  $h$ .

Soient  $P$  un sommet libre<sup>2</sup> et  $(K_i)$  l'ensemble des triangles contenant  $P$  comme sommet. Ce dernier constitue un polygone étoilé par rapport à  $P$ . Le bougé du point  $P$  consiste alors à déplacer  $P$  pour augmenter la qualité des triangles  $K_i$  (en particulier la plus mauvaise); une conséquence immédiate de ce fait est alors la diminution des variations trop importantes des  $h$ . Deux méthodes ont été envisagées : barycentrage et bougé optimal.

#### 7.1.1 Barycentrage

Soient  $(Q_j)$ , les sommets des triangles  $t_i$ , autre que le point  $P$ . La méthode de barycentrage consiste à remplacer  $P$  par le barycentre des  $Q_j$  si la qualité la plus mauvaise

<sup>2</sup>Un sommet est dit libre s'il n'est pas sur la frontière du domaine, s'il n'est pas sur un contour interne ou encore s'il n'est pas spécifié explicitement en tant que point fixe.

des triangles  $K_i$  est améliorée (figure 17). Ce procédé est appliqué à tous les sommets libres et est itéré plusieurs fois (en pratique 3 itérations sont suffisantes). Le processus est convergent car le barycentrage traduit entre autre un laplacien.

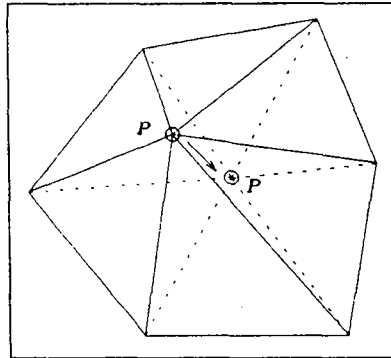


FIG. 17 - *Barycentrage.*

### 7.1.2 Bougé optimal

Soient  $(F_i)$ , les arêtes frontières du polygone formé par les triangles  $(K_i)$ . Pour chaque arête  $F_i$ , on définit un point "idéal"  $P_i^*$ , pour que le triangle  $\{F_i, P_i^*\}$  soit équilatéral et que ce dernier soit du même côté de  $F_i$  par rapport au triangle  $\{F_i, P\}$ . Dans ce cas, le bougé consiste à déplacer "pas à pas" le point  $P$ , vers le barycentre des points  $(P_i^*)$ , si la qualité la plus mauvaise des triangles  $K_i$  est améliorée (figure 18).

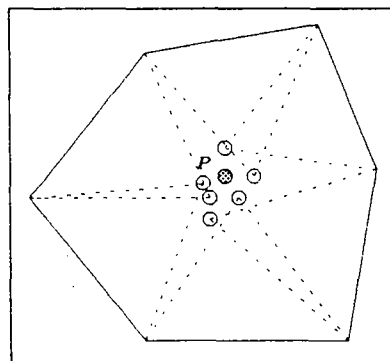


FIG. 18 - *Bougé optimal.*

**Remarque.** On peut appliquer d'autres variantes de ces procédés en considérant des barycentres pondérés [4].

## 7.2 Optimisation

L'optimisation n'est que le retournement d'arête, effectué dans le cas où la qualité des éléments obtenus est meilleure.

## 8 Schéma du mailleur

Il reste à regrouper les résultats des sections précédentes pour obtenir un algorithme de maillage capable de traiter tout domaine, convexe ou non, de frontière non croisée.

Le schéma proposé est le suivant :

### Schéma du mailleur.

- a) Création du nuage de points associé aux données, c'est-à-dire les points des arêtes du contour du domaine et les points spécifiés;
- b) Détermination de la position de quatre points supplémentaires tels que le quadrilatère formé par ceux-ci englobe tous les points du nuage;
- c) Maillage de ce quadrilatère à l'aide de deux triangles;
- d) Insertion, un à un, des points du nuage pour construire un maillage comprenant comme sommets d'éléments ces points;
- e) Recherche dans  $T$ , le maillage précédent, des arêtes du contour donné:
  - e1)  $T$  contient les arêtes données, aller en f);
  - e2) Sinon, modifier localement  $T$  en recréant, une à une, les arêtes manquantes;
- f) Marquage des triangles de  $T$  en fonction de leur position vis-à-vis du contour;
- g) Création des points internes et insertion de ceux-ci;
- h) Elimination des triangles extérieurs;
- i) Régularisation et optimisation éventuelles.

**Remarques.** Les étapes a) b) et c) (tout comme le fait de n'éliminer l'extérieur qu'en h)) permettent d'être constamment dans un cas convexe. L'insertion des points (étapes d) et g)) est faite via l'algorithme 6, ceci donne le nom de la méthode (et le titre du papier).

### 8.1 Quelques exemples

Sont donnés maintenant quelques exemples de maillages obtenus en employant cette méthode. Pour chaque exemple, on donnera le nombre de triangles, le nombre de points, le temps cpu (HP 735) en secondes, la qualité du résultat c'est-à-dire la répartition des

triangles en fonction de leur forme mesurée via la quantité  $Q = \sqrt{3} \frac{\rho}{d}$  ( $\sqrt{3}$  est le coefficient de normalisation tel que  $Q = 1$  pour un triangle équilatéral) où  $d$  est la longueur de la plus grande arête de l'élément mesuré,  $\rho$  est le rayon de son cercle inscrit (notons que  $0 < Q \leq 1$  diminue quand le triangle s'aplatit) et la variation en moyenne de  $h$  en chaque point interne généré (pour un tel point,  $h$  est la moyenne arithmétique des longueurs des arêtes issues de ce point).

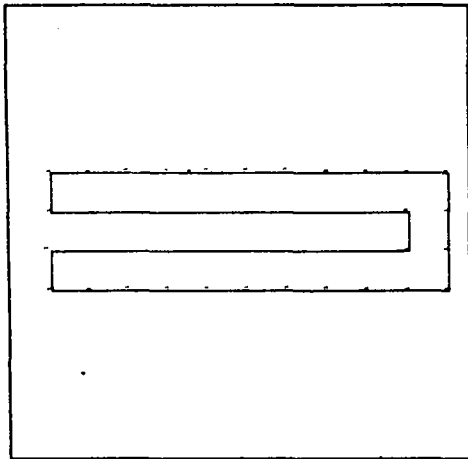


FIG. 19 - "porc-épic": la donnée.

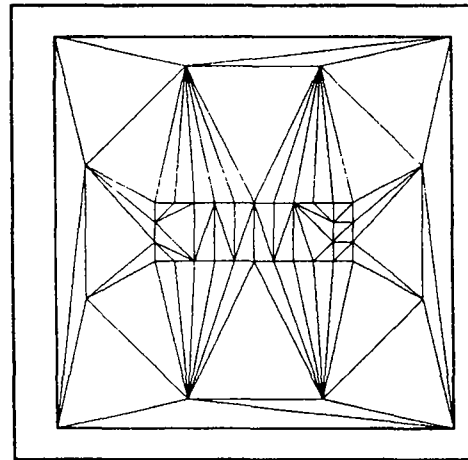


FIG. 20 - "porc-épic": le maillage englobant.

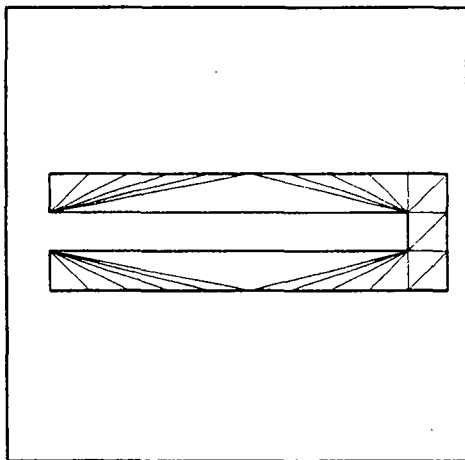


FIG. 21 - Résultat.

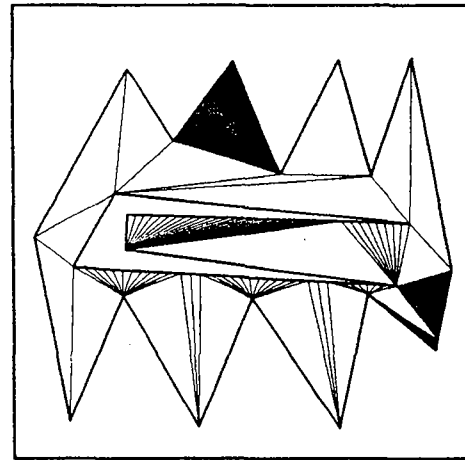
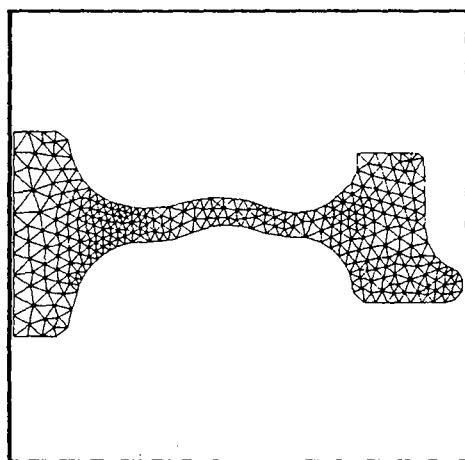
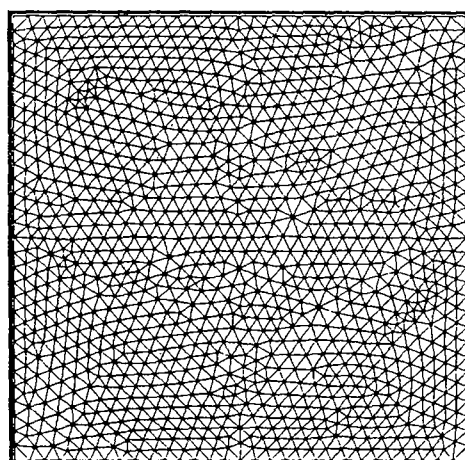


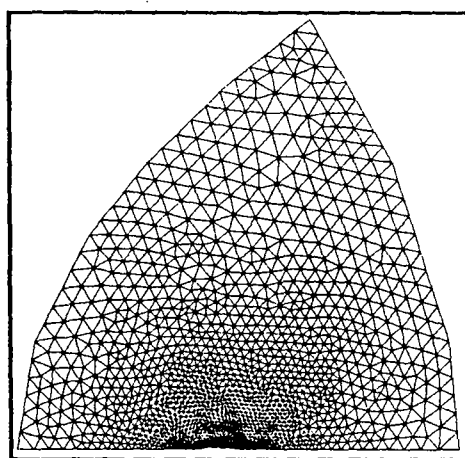
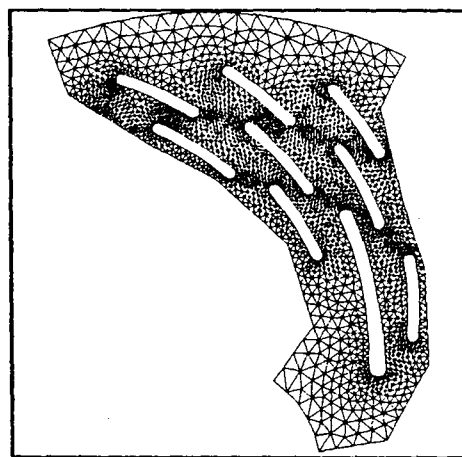
FIG. 22 - Autre exemple de test.

La figure 19 montre un exemple de contour "mal" discrétisé produisant la non création de ses arêtes dans le maillage du quadrilatère englobant (figure 20). La phase de régénération de la frontière est donc utilisée pour construire les arêtes manquantes et obtenir le résultat (figure 21). L'exemple montré sur la figure 22 a été construit pour tester cette phase de recréation d'arêtes manquantes, le test n'est en aucun cas réaliste, mais il illustre la robustesse de la méthode.

La figure 23 (cas 1 du tableau 1) montre l'exemple "habituel" utilisé dans Modulef qui est celui d'une coupe d'une roue. La figure 24 (cas 2) illustre le cas d'un carré et,

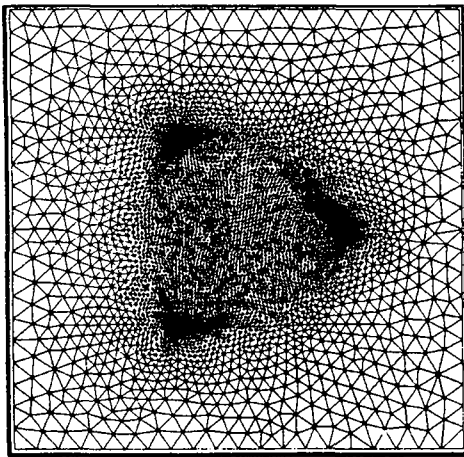
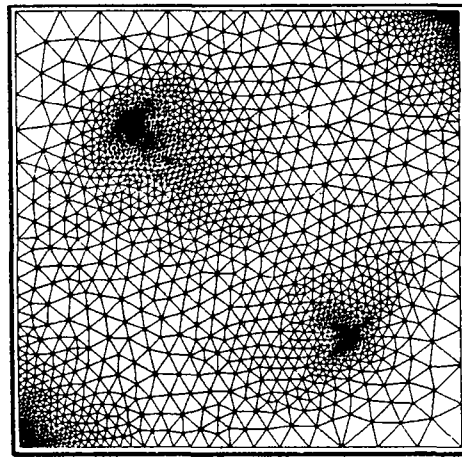
FIG. 23 - *La coupe d'une roue.*FIG. 24 - *Le cas d'un carré.*

contrairement à ce que l'on pourrait croire a priori, nous paraît plus significative du comportement de la méthode au sens où une géométrie simple n'est pas généralement correctement maillée par un mailleur automatique.

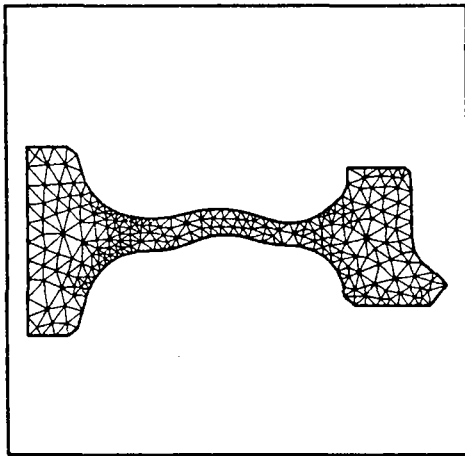
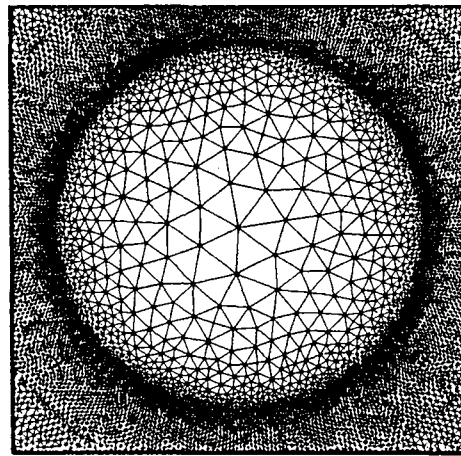
FIG. 25 - *Exemple de profil.*FIG. 26 - *Cas d'une simulation sur un frein.*

La figure 25 (cas 3 du tableau 1) montre l'exemple classique d'un profil, la figure 26 (cas 4) correspond à une simulation thermique sur un disque de frein et les figures 27 et 28 (cas 5 et 6) le cas d'un carré défini par son contour, discrétisé à l'aide de progressions constantes et variables, et quelques points internes spécifiés (avec des  $h$  petits).

Par ailleurs, un paramètre  $\omega$  a été introduit pour diluer l'influence des  $h$  frontières au fur et à mesure que l'on s'éloigne de celle-ci ( $\omega \geq 1$ , une valeur 1. donne un résultat neutre, une valeur 1.2 indique une grande dilution et diminue le temps de calcul nécessaire). La figure 29 reprend le cas de la roue déjà vue pour un paramètre  $\omega = 1.2$ , dans ce cas la différence est faible car la dilution est une fonction de la distance entre un point et la frontière et de ce fait n'est jamais importante. La figure 30 illustre le cas d'un cercle, cas dans lequel, en diminuant la taille des segments discrétisant le contour, on augmente la

FIG. 27 - *Cas des points internes spécifiés.*FIG. 28 - *Autre cas de points internes imposés.*

distance relative entre un point interne (par exemple le centre de gravité) et la frontière.

FIG. 29 - *Influence du paramètre  $\omega$ .*FIG. 30 - *Influence du paramètre  $\omega$ .*

Cas de test	1	2	3	4	5	6
Nombre de triangles	593	2272	3339	5680	12740	9424
Nombre de sommets	356	1197	1757	2916	6411	4847
Temps cpu (en sec.)	0.07	0.26	0.41	0.66	1.8	2.18
Qualité	0.45	0.52	0.26	0.41	0.45	0.45
Nombre de triangles "mauvais"	3	0	11	11	14	28
Qualité après régularisation	0.6	0.66	0.35	0.54	0.6	0.55
Nombre de triangles "mauvais"	0	0	5	0	0	0
variation moyenne de $h$ en %	11	6	10	12	10	14

Tableau 1 : *Résultats relatifs aux maillages ci-dessus.*

Le tableau 1 donne comme qualité du maillage, la plus mauvaise valeur atteinte et indique combien de triangles sont de mesure plus mauvaise que 0.5. Dans l'implantation, on peut effectuer une régularisation dans le cas où le résultat brut possède au moins un triangle dont la qualité est plus petite que 0.5 (ceci augmente légèrement le temps et correspond aux dernières lignes du tableau).

Cas de test	1	2	3	4	5	6
Nombre de triangles	986	3988	15602	62094	249174	991232
Nombre de sommets	534	2075	7962	31368	125228	496597
Temps cpu (en sec.)	0.12	0.43	1.79	7.3	33.3	158

Tableau 2: *Temps requis en fonction de la taille du maillage.*

Le tableau 2 donne le temps nécessaire en fonction du nombre de triangles créés. L'exemple pris est le carré déjà utilisé pour lequel on fait varier le nombre de points sur le contour (notons que pour cet exemple, la qualité la plus mauvaise est toujours de l'ordre de 0.65).

Calculer la vitesse théorique de la méthode relève d'un exercice de style. Il nous paraît plus informatif de regarder les résultats obtenus (cf. tableau 2) qui donnent une idée des performances atteintes effectivement. Notons d'autre part qu'un calcul de complexité ne donnerait qu'une valeur biaisée puisqu'il ne s'appliquerait qu'au processus d'insertion de points, sans tenir compte des difficultés numériques (erreurs d'arrondis) et sans prendre en compte la nécessité de préserver les arêtes initialement données.

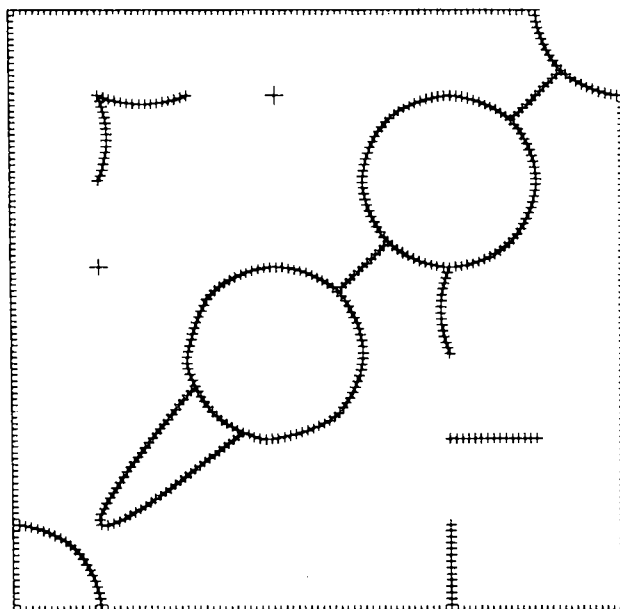
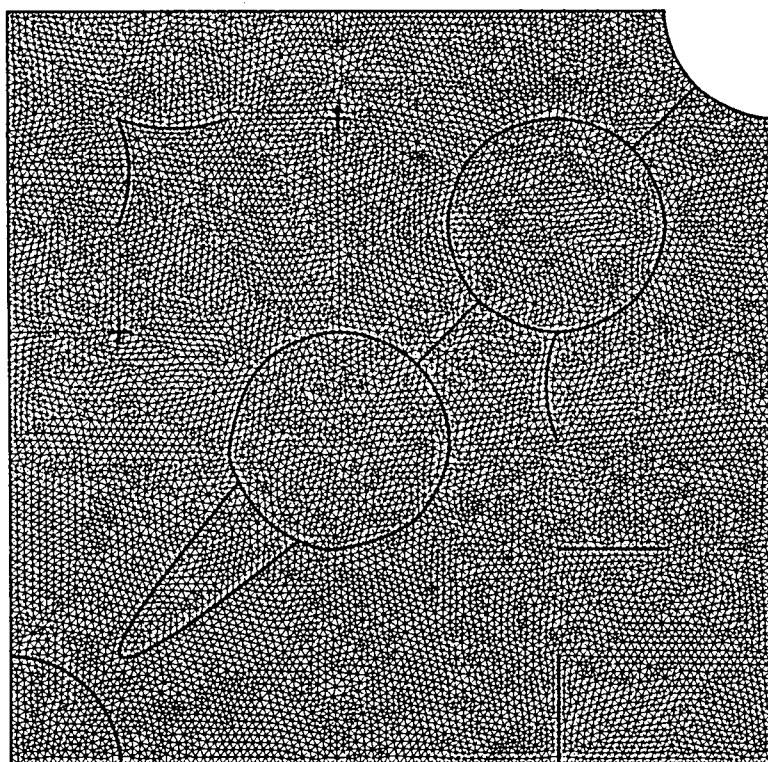
Enfin on considère un dernier exemple préparé par le logiciel de maillage 1D de P.Laug [16], dans lequel des points, des arêtes, des contours et des sous domaines internes sont imposés. Dans un premier temps, la frontière du domaine, les arêtes, les contours et la frontière des sous domaines sont discrétisés à  $h$  constant (figure 31). Nous obtenons alors un maillage "presque uniforme" (figure 32) constitué de 23542 triangles. Puis, on considère une discrétisation à  $h$  variable (figure 33), ce qui donne le maillage de la figure 34 qui comprend 29948 triangles.

On constate alors que dans les deux cas la qualité la plus mauvaise est de l'ordre de 0.55 et que la variation en moyenne de  $h$  est de l'ordre de 8%. On en déduit que le maillage 1D du domaine (pourvu que la variation de  $h$  ne soit pas trop important) n'a pas d'influence significative, en terme de qualité et de variation de  $h$ , sur le maillage 2D.

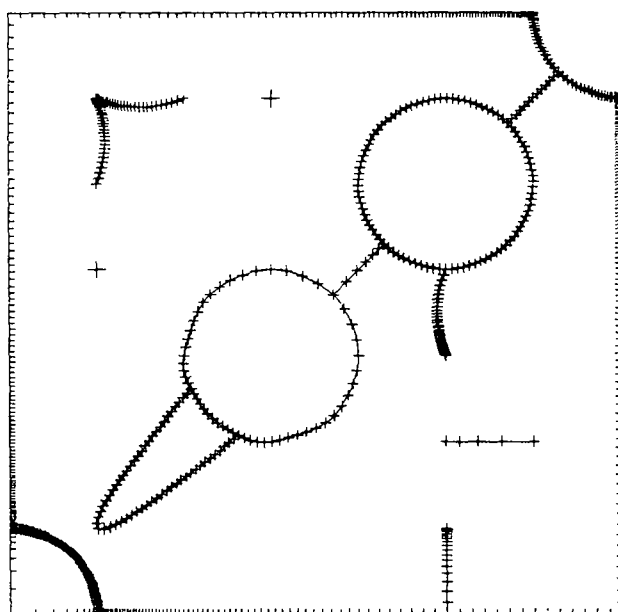
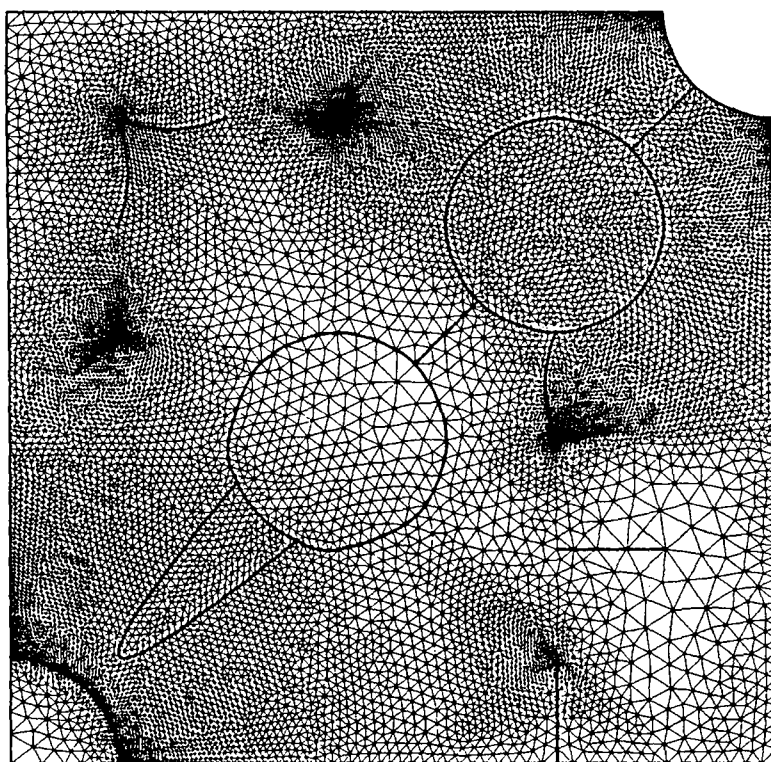
## 8.2 Conclusion

En conclusion, le mailleur décrit semble numériquement fiable, rapide et donne des maillages de bonne qualité. Les points internes sont constitués par paquet, ce qui permet d'exploiter au maximum l'effet positif d'une randomisation dans l'insertion. Ceci est une différence notable avec la méthode développée dans [14] et, nous semble t-il, constitue un point fort de la méthode. Il reste à étendre les résultats à la dimension 3 (cf. ci-dessous pour les principales idées relatives à ce cas<sup>3</sup>).

<sup>3</sup>En fait, le vrai problème est celui de la dimension 3 et cette étude n'a pour but que de permettre une présentation facile à voir des idées.

FIG. 31 - *discrétisation à  $h$  constant.*FIG. 32 - *maillage à  $h$  constant.*



FIG. 33 - *discrétisation à  $h$  variable.*FIG. 34 - *maillage à  $h$  variable.*

## 9 Application à la dimension 3

Nous allons, dans cette courte section, tenter de recenser les extensions possibles des résultats présents au cas de la dimension 3. Nous reprenons l'ordre dans lequel ils apparaissent.

- Outils : la notion de triangle est remplacée par celle de tétraèdres définis par la liste orientée de 4 sommets, possédant 4 voisins par face (la notion de face remplace la notion d'arête).  $B(K)$  est la boule ouverte circonscrite à l'élément  $K$ .
- Normalisation : le souci est de tenir exact non plus des surfaces mais des volumes (i.e. une longueur au cube).
- Position d'un point : l'algorithme 1 se généralise sans difficulté.
- Recherche de l'élément contenant un point  $P$  donné : l'algorithme 2 se généralise sans difficulté.
- Éléments coupés par un segment : l'algorithme 3 se généralise.
- Éléments coupés par un triangle : point nouveau spécifique à la dimension 3.
- Retournement d'arêtes : sa généralisation conduit à définir des opérateurs nouveaux portant, en particulier, sur la manipulation d'ensemble nommé coquille [4].
- Procédures d'accélération : la grille et les transports des centres se généralisent sans problème, par contre les relations de voisinage peut être étendues moyennant une étude plus fine.
- Insertion de points : la méthode s'étend sans problème.
- Respect d'une frontière : l'extension au cas 3D reste un point délicat (voir [10] et [11]).
- Marquage de l'extérieur : l'algorithme 8 convient dans son esprit.
- Points internes : la méthode algébrique peut être étendue aisement.

La plupart des propositions a été expérimentée et intégrée dans une nouvelle version du logiciel GHS3D.

## Références

- [1] M.BERNADOU ET ALL, *Modulef : une bibliothèque modulaire d'éléments finis*, INRIA, 1988.
- [2] J-D.BOISSONNAT AND M.TEILLAUD, *A hierarchical representation of objects : The Delaunay Tree*, Second ACM Symp. on Comput. Geom. in Yorktown Heights, 1986.
- [3] H.BOROUCHAKI ET P.L.GEORGE, *Encore Delaunay*, Rapport de Recherche à paraître, INRIA, 1995.

- [4] E.BRIÈRE DE L'ISLE ET P.L.GEORGE, *Optimisation de maillages tridimensionnels*, Rapport de Recherche n° 2046, INRIA, 1993.
- [5] C.CHERFILS, F.HERMELINE, *Diagonal swap procedures and characterizations of 2D-Delaunay triangulations*, M<sup>2</sup>AN, vol 24, n° 5, pp. 613-625, 1990.
- [6] B.DELAUNAY, *Sur la sphère vide*, Izv. Akad. Nauk SSSR Otdelenie Matemat. Estestvennyka Nauk, Vol 7, pp.793-800, 1934.
- [7] P.L.GEORGE, *Génération automatique de maillages. Applications aux méthodes d'éléments finis*, collection RMA n° 16, Masson, Paris, 1990.
- [8] P.L.GEORGE, *Modulef: Construction et modification de maillages*, Rapport Technique n° 104, INRIA, 1989.
- [9] P.L.GEORGE, *Génération de maillages par une méthode de type Voronoï*, Rapport de recherche n° 1398, INRIA, 1991.
- [10] P.L.GEORGE, F.HECHT, E.SALTEL, *Fully automatic mesh generator for 3D domains of any shape*, Impact of Computing in Science and Engineering, vol 2, pp. 187-218, 1990.
- [11] P.L.GEORGE, F.HECHT, E.SALTEL, *Automatic mesh generator with specified boundary*, Comp. Meth. in Appl. Mech and Eng., 92, pp. 269-288, 1991.
- [12] P.L.GEORGE, F.HERMELINE, *Delaunay's mesh of a convex polyhedron in dimension d. Application for arbitrary polyhedra*, Int. J. Num. Meth. Ing., vol 33, pp. 975-995, 1992.
- [13] F.HECHT, E.SALTEL, *Emc2: Editeur de maillages et de contours bidimensionnels*, Manuel d'utilisation, Rapport Technique n° 118, INRIA, 1990.
- [14] F.HERMELINE, *Une méthode automatique de maillage en dimension n*, Thèse, Université Paris 6, Paris, 1980.
- [15] F.HERMELINE, *Triangulation automatique d'un polyèdre en dimension N*, Rairo, Analyse numérique, vol 16, n° 3, pp. 211-242, 1982.
- [16] P.LAUG, *Structuration des données dans les calculs par éléments finis*, Strucome 94, 1994.
- [17] A.PERRONNET, *A generator of tetrahedral finite elements for multi-material object and fluids*, Numerical grid generation in computational fluid mechanics'88, Miami, 1988.
- [18] F.P.PREPARATA, M.I.SHAMOS, *Computational geometry, an introduction*, Springer-Verlag, 1985.
- [19] J.Y.TALON, *Algorithmes de génération et d'amélioration de maillages en 2D*, Rapport Technique n° 20, Artemis-Imag, 1987.
- [20] M.G.VALLET, *Génération de maillages anisotropes adaptés. Application à la capture de couches limites*, Rapport de Recherches n° 1360, INRIA, 1991.
- [21] D.F.WATSON, *Computing the n-dimensional Delaunay Tessellation with applications to Voronoi polytopes*, Computer Journal 24, n° 2, pp. 167-172, 1981.

- 
- [22] N.P.WEATHERILL, *The generation of unstructured grids using Dirichlet tessellation*, MAE report n° 1715, Princeton Univ., 1985.
  - [23] N.P.WEATHERILL, *The integrity of geometrical boundaries in the 2-dimensional Delaunay triangulation*, Comm. in Appl. Num. Meth., vol 6, pp. 101-109, 1990.



---

Unité de recherche INRIA Rocquencourt  
Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)  
Unité de recherche INRIA Lorraine - Technopôle de Nancy-Brabois - Campus scientifique  
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)  
Unité de recherche INRIA Rennes - IRISA, Campus universitaire de Beaulieu 35042 Rennes Cedex (France)  
Unité de recherche INRIA Rhône-Alpes 46, avenue Félix Viallet - 38031 Grenoble Cedex 1 (France)  
Unité de recherche INRIA Sophia Antipolis - 2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

---

Éditeur  
INRIA - Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)

ISSN 0249 - 6399



★ R R - 2 4 6 3 ★